

BEEBLET

£3-00

ISSN 0112-3351



CONTENTS

Editorial	2
President's Report	3
Classified Ads	4
Regional News	5
HELP	6
Introducing SAS	7
Short listings	10
Review: MUSIC500	13
Video Production	17
Getting Iffy	22
File Dump	25
Memories are made of this	30
Review: Creative Assembler	34
GRID	36
A letter	37
Hall of Fame	39
Local Groups	40

VOLUME 3

NUMBER 4

MAY 1985



BBC/ACORN
COMPUTER USER GROUP
NZ (inc)



EDITORIAL

Innovation! In response to a number of suggestions (survey, letters to the Editor, casual comments) I have introduced two changes to this issue. First, the contents on the outer cover (and you are welcome to send in new cover designs). Second, wider character spacing (and therefore less material), on the pages with a solid block top and bottom. Some old format pages (carried over from April) remain for comparison. Please send in your opinions.

CONTRIBUTIONS

I think perhaps the focus of this debate could be usefully switched to just what the newsletter, and the User Group, are all about. To keep the former viable in its present form we do need a high level of member participation. However UG members have a very wide range of interests and requirements, often conflicting. Both the group and the newsletter should reflect and adapt to this. The National Committee is certainly looking very closely at the survey responses, and any other communications on the subject. Keep them coming in!

COMPETITION - \$50 Software prize(s)

The COMPETITION I announced last month is deliberately wide open. This is to encourage people at all levels and interests to participate. By well written I mean well thought out, not written by an "expert". I will be looking at factors such as originality, usefulness and entertainment value where relevant. Size is not particularly important (function key definitions and procedures will be considered, especially if in sets.)

ELECTRON SOFTWARE

All 1984 and 1985 'Beeblet' programs have been tested on an Electron as well as the BBC. Almost all work without change. All Electron owning members have been sent a list of the programs, commented as to level of compatibility. Anyone wanting a copy of the list should send a SAE to the UG (Box 9592, Wellington).

SUGGESTIONS, REQUESTS

I am working my way through the suggestions and queries in the survey and other recent correspondence. This will be reflected in future Beeblets. Keep the ideas coming.

PRESIDENT'S REPORT - MAY 1985

This is the first of what is intended to be a regular report to members on the state of your User Group, activities of the National Committee, and anything else that may be appropriate.

As of April, the User Group had about 450 members. Among the many benefits of membership are being able to meet, learn from, and help other enthusiasts, to save money by participating in bulk deals, and in particular to receive this monthly newsletter. Those in more remote areas have limited opportunities to meet directly with others, and must keep in touch with activities going on around the country through the newsletter. Those in larger centres do have local meetings, and should be taking advantage of the opportunity to use "Beeblet" to publicize their activities and give notice of forthcoming events. All members benefit by articles explaining the various features of the BBC and Electron microcomputers, programming techniques, news of what Acorn and Barsons are up to, program listings, reviews (from a NZ point of view), and so on.

At the AGM, there was considerable discussion about the level of subscriptions and whether "Beeblet" should perhaps be reduced to a bi-monthly publication or included as an insert to another magazine such as "Bits and Bytes". To some extent, any dissatisfaction appeared to arise from feeling that the magazine was not providing value for money, rather than that the subscription amount was too much. The Committee has been directed by the AGM to canvass members (and non-members) on the alternatives. At our first Committee meeting in April we did not have time to consider this in detail, but as a first step we will be carefully reading completed survey forms included with the membership renewals. As well, a number of steps will be taken to improve "Beeblet" - the first has been the re-introduction of the "Help" column as of last month.

I, like a lot of you, have saved the cost of my subscriptions many times over through bulk deals arranged by the User Group. While the National Committee dealt in cassettes, disks, and cables, the larger items such as printers and disk drives were handled by the Wellington local branch, via the extraordinarily capable dealing of Jill Armstrong. Jill has now stepped down from the Wellington committee and, in lieu of a successor, the job of handling the larger deals has been passed to the National Committee. While meaning more work for us, or more particularly for the Equipment Officer (Keith O'Callahan), it does make the handling of orders and money much more tidy and straightforward.

Finally, a word on the modem project. Slippage. Without a vital part (the crystal) it was not possible to get a prototype assembled in time for the Wellington meeting at the end of April. The crystal is now in and the modem will be tested over the next couple of weeks, then a new prototype built in printed circuit board form. This will be submitted for type approval as a direct connect modem. The price (as a kitset) still looks like being around \$150.

Neil Gordon

CLASSIFIED ADS

DEALS AND DISCOUNTS

Send stamped addressed envelope for details of the deals negotiated by the Wellington Branch for UG members. Please mark your request for the attention of Keith O'Callahan, C/O the UG Box 9592 Wgtn.

FOR SALE:

MICRO COVERS

Designed to fit the BBC micro. Attractive opaque reinforced plastic, \$5-50 posted. To Iain, Box 26078, Epsom, Auckland 3.

CASSETTE TAPES

C10s, boxed, leaderless - available from the UG in multiples of 20 : \$26-00 plus \$2-00 p&p.

FLOPPY DISCS

3M Brand 5 1/4" DSDD discs - available from the UG ex-stock in (cardboard) boxes of 10: \$55-00, which includes p&p.

PRINTER

10" Dot matrix printer :STAR DP510, good quality and reliable. \$500-00. Machine-code screen dump included. Contact David Fung, 2 Glen Road, Kelburn, Wellington.

BBC B

BBC Model B, O.S. 1.2, BASIC 1. Zenith green screen monitor, cassette recorder. Any offers? Phone Wellington 735-230 after 7 pm. Ask for Duncan.

DISC DRIVE

Sale/Swap: I have a Shugart disc drive and a Mitsubishi disc drive and I want two of the same. I will swap either one for the other type or will sell the Shugart for \$480-00 ono. Phone Jeff Hunt 795-835 Wellington.

REGIONAL NEWS

AUCKLAND: Steve Williams writes, "Last meeting was communications and Terry Ballard from Datacomm came down and gave a talk on the technical side which went about a mile over most people's head! Very good talk though... Innovation was a beginner's corner chaired by Mike Nudds, very well received.

Next meeting is Dave Smith talking about the use of a BBC in Share performance analysis. He has a business which specialises in investment consultancy so I guess it must work...

Upcoming meeting topics:

CAVII videotape interface
Expansion options (2nd processor, SOLIDISK, ARIES etc..)
Doug Pauling from Barson's (topic unnecessary!)
whatever anyone else suggests..

I have taken over as editor of the BBC pages on ADITEL. Look at pages 60025 down. All contributions gladly accepted.. (Now, where have I heard that before?)

WELLINGTON: Last meeting (Wednesday 24 April) was the A.G.M. A Committee was duly elected.

Programme:

29 May: Printers
25 June: Spreadsheets
31 July: Modems
28 August: Radio
25 September: Music & Sound

There is also a workshop being planned for mid-June.

CHRISTCHURCH: PROGRAMME-

27 May: Good Programming Techniques I
10 June: " " " II
24 June: Guest Speaker T.B.A.
8 July: Open forum.

DUNEDIN: Martin McDowall reports "We had a meeting on Thursday night the 7th (of March) with a pleasing turn out of BBC and Electron people.. We have probably have more of an interest come User's Group thing at this stage but most seemed pleased they came and intend to come again.." Contact Martin for further details (see back cover).

HELP !

A forum conducted by Tim Ryan

Steve S who is an Electron owner asks:

Q1 Can a picture and/or text be saved on video for viewing on any video machine?

A1 Sure ! All you do is connect the RF output of your Beeb to the aerial input of the video unit using the standard cable that connects to a TV, then put the picture up on the Electron's screen and record away. At one time I was experimenting, and recorded myself playing Space Invaders. I was then able to play back the video one frame at a time and see exactly what was happening

Q2 Can an Electron act as a teletext adapter to save having to buy a TV with teletext built in ?

A2 No I'm afraid not. Even the BBC with it's teletext Mode can't do this as it requires a piece of special circuitry to pick the teletext screens off the TV signal.

Dave F asks

Q3 How do you centre a machine code dump when set-margin and TAB commands to the printer do not work ?

A3 With difficulty. Actually it's quite simple as long as you are able to write machine code. What you need to do is insert a little routine just before each line is printed to move the carriage to the right. Eg:

```

LDX n
LDA ASC" "
centre JSR OSWRCH
DEX
BNE centre
```

All this does is print "n" spaces before each lot of graphics. Note this must be done BEFORE the printer is sent the control codes for graphics.

Further to my answer (A5) in last month's magazine about programs being run at &E00, neither LOMEM nor TOP are reset unless you do an OLD. Therefore you must do an OLD as the last part of your move down routine.

Q4 One problem often found when typing in programs from magazines is the error "No such variable at line XXX"

A4 The usual reason for this is that you have typed the variable name wrong. There are two places at which this may have happened:

1. At the line pointed to by the error message.
 2. At the line where the variable is first given a value.
- Eg. If you've put

```
demoliton=0
result=demolition+i
```

Then BASIC will point out that you have no such variable in the second line even though the error is in fact in the first line.

Neil B asks about the random number generator (RNG):

Q5 Can the generator be made truly random ?

A5 Not easily at all, unfortunately. In fact if you could come up with a true RNG I bet the scientists would beat a path to your door !

Probably the best you can do is to use TIME and manipulate the values obtained to provide a random number within the required range. Unfortunately this means you can't get a random number very often, so another way to do it is to use TIME to seed the RNG by putting dummy=RND(-TIME).

The usual way to get a true random number is to use a "noisy" component such as a diode, and read off the voltage, or current across it. This, though, does not give a very good distribution of values, and would be difficult to implement on the BBC.

Hint: It is most important to get exactly the right number of values for a VDU command. If you don't, anything sent through the VDU vector afterwards (Eg PRINT, VDU, MOVE, PLOT) will be used to complete the original command.

Hint: Always check for alternative ways to solve your problem. The first idea you come up with may well not be the best solution.

INTRODUCING SAS

G.C.F.

PREAMBLE:

This is part of a series of articles on adventure interpreter programs. The overall project was outlined in the March '85 'Beeblet', and the first adventure program (VSAS) appeared in April.

SAS, here, refers to Simple Adventure System. It does not relate to the Special Air Service's regiment of crack troops, nor to any other software with similar initials. As SAS is somewhat more involved than VSAS, it will be spread over several issues (There is more to life than this apparently!). The articles will concentrate on how to use SAS to write your own adventures. Hence only brief details will be given of the logic and coding techniques of the interpreter itself.

PROGRESS

For more than 3 months I have been labouring over SAS, without actually writing any code (ie a BASIC program) apart from a couple of PROCs to establish the feasibility of certain coding techniques - and they were only on paper!

I have done the 11th revision, and have ended up with 7 A4 pages of design notes, and over 20 pages detailing a sample adventure for it. If the current schedule can be maintained, a working (de-bugged but not polished) program should be finished mid-June. In any project of reasonable size, less than 20% of the time consists of coding, debugging and polishing. 79% of the time is spent on thorough analysis and design. The remaining 1%?....

FEEDBACK

DON A (Tokoroa) suggests that VSAS and SAS have educational possibilities (he teaches foreign languages - "prenez la lumiere"? [take the lamp])

This is very true! SAS can be thought of also as an enhanced multi-choice program where the answer to one set of questions not only decides the next set, but also share a common theme:

Building a house - first set of questions concern location constraints: eg if near swamp, a later set deals with what you will do when there is a flood?

JAMIE D (Westport) suggests that graphics scenes could be stored on disc and brought in when required (presumably via *LOAD).

This is a good idea which could easily be incorporated into the system; it may require separate text and graphics windows. Most of the adventure code would also have to be accessed from disc, rather than from DATA, to save space (unless you have a co-processor or sideways RAM) - this is starting to look like VAS

TIM R (Petone) suggests using two programs. The first containing DATA, which would then be compacted into memory. The second would access this by peeks and pokes (indirection operators ?*!). This would certainly make better use of memory (cf FAS)

SEVERAL have suggested writing it in machine code for "speed" and compactness. Fine if you have the time and competence, but I find machine code too slow to write and debug - besides how do you duplicate the EVAL function, and implement Recursion? Both these are needed for SAS.

If you can answer these two, please put it into an article in Beeblet. Don't expect me to - I haven't yet had sufficient justification to write ANY machine code (yet) for the Beeb!

CONCLUSION

If I don't immediately implement any of the above, it is not because they lack merit, but because there are other demands on my time (this weekend for instance, I have to write this article, change nappies, help my wife design, write and debug another program by Monday, eat, sleep, enjoy the family,...) What's Beeblet for if I can't pass the programming buck to some of you?

ADVENTURE DESIGN POINT

This type of adventure is well suited to adventures with lots of rooms with complicated descriptions and functions, but with a very small number of objects to actually be picked up. Shades of crosswords? Cryptic clues?

If the adventure you want to write fits into the above description, then you will be as amazed as I was at the level of sophistication possible (and I've been designing the system!).

SHORT LISTINGS

Neil B. has supplied a number of short interesting programs. Here are a few.

GRAVITY:

This calculates the gravitational force of attraction, acceleration and relative velocity versus time for two spherical (ideally) objects floating freely in space, with zero initial relative velocity and without interference from other gravitational fields. The user inputs the increment of time which the program uses to calculate the conditions for each iteration. The smaller the increment the greater the number of calculations made before impact and the higher the accuracy of the resulting data (especially that for acceleration and velocity over great distances). If you find that your time increment is so small that nothing seems to be happening, press the F (faster) key will multiply the increment by 10 and restart the process. If the increment selected is too great (or you pushed F too many times) the calculations will rerun at one tenth speed to give at least 40 iterations for a better result.

The formulas are based on Newton's laws and work for a very wide range of values. As a test, try jumping from a height of 10 metres to find out how hard you hit the ground etc. (function 1 and 2 are programmed with the earth's mass and radius respectively - take combined radii as equal to the Earth's radius). The gravitational effect of your puny body on the Earths is quite revealing!

MISSILE

Again, based on simple Newtonian formulae, graphs the flight of an object thrown at various angles above the horizontal. N.B. it takes no account of air resistance, curvature of the earth or reduction of gravity with height above the Earth, so will give quite misleading information about large velocities!

TWINKLE

Based on listing for another machine.

This produces random star constellations, and colours used were selected to give three levels of magnitude on a green screen, which gives a very realistic effect in a dark room with the screen brightness turned down low. Experiment with other colours for a colour screen. Note that the pattern changes gradually as stars are extinguished and new ones created.

```

10 REM::GRAVITY::Creative Computing
12/83
20 REM ADAPTED, CORRECTED, STRUCTURED
AND EXPANDED (with GRAPHICS) NEIL B.
2/85
30 *K.0 RUNIM
40 *K.1 5.98E24 REM MASS OF EARTH (
kg)
50 *K.2 6.375E6 REM RADIUS OF EARTH
(metres)
60 MODE0
70 PRINTTAB(23,2)"G R A V I T Y   E
X P E R I M E N T"
80 INPUT""Enter Initial distance
between object surfaces, in metres ",Dini
90 REPEAT
100 INPUT"Enter smaller Mass (kg)
",m
110 INPUT"Enter larger Mass (kg)
",M
120 UNTIL M>=m
130 INPUT"Enter calculation time in
crement (secs) ",DT
140 INPUT"Enter combined radii of o
bjects (m) ",comrad
150 R=Dinit+comrad:Rinit=R
160 gravcon=6.664E-11
170 escvel=SQR(2*gravcon*M/R)
180 PROCreset
190 VDU5
200 REPEAT
210 REPEAT
220 PROCiterate
230 UNTIL R=comrad
240 IF Count<41 DT=DT/10:PROCreset
250 UNTIL Count>40
260 FORX=-13 TO-3 STEP2
270 SOUND0,X,4,4
280 NEXT
290 CLS
300 VDU4
310 PRINTTAB(30,2)"R E S U L T   D
A T A"
320 PRINT""Small mass: ",m" kg"
330 PRINT"Large mass: ",M" kg"
340 PRINT"Initial separation: ",Dini
t" m"
350 PRINT"Time calculation unit used
: ",DT" secs"
360 PRINT"Combined radii of objects:
",comrad"m"
370 PRINT"Initial velocity req'd for
small object to escape large one: ",jes
cvel" m/s"
380 PRINT"Final acceleration due to
larger mass = ",acc" m/s/s"
390 PRINT""Time to collision: ",Cou
nt*DT" secs or ",Count*DT/3600" hours."
400 IF Count*DT/3600/24 PRINT"That i
s ",Count*DT/3600/24 " days!"
410 IF Count*DT/3600/24>365 PRINT"Th
at is ",Count*DT/3600/24/365 " years!!
"
420 PRINT"Velocities at impact:"
430 PRINT "small object ",v" m/s"
440 PRINT "large object ",V" m/s"
450 PRINT""Distances travelled:"
460 PRINT "small object ",1/(OSCLIOS
CLIL)*Dinit" m""large object ",L/(1+L)
*Dinit" m"
470 END
480
490 DEFPROCiterate
500 Count=Count+1
510 force=(m/R)*(M/R)*gravcon
520 acc=force/m:ACC=force/M
530 v=v+acc*DT:V=V+ACC*DT
540 d=acc*(DT*DT)/2+(v*DT):D=ACC*(DT
*DT)/2+(V*DT)
550 R=R-d-D
560 l=l+d:L=L+D
570 MOVE 0,500:DRAW 1*1275/Dinit,500
580 MOVE 1275,500:DRAW 1275-(L*1275/
Dinit),500
590 A$=INKEY$(0):IF A$="F" DT=DT*10:
PROCreset:REM SPEEDS THINGS UP IF MOVEM
ENT TOO SLOW
600 ENDPROC
610
620 DEFPROCreset
630 CLS:SOUND1,-12,150,2
640 Count=0
650 R=Rinit
660 v=0:V=0
670 l=0:L=0
680 ENDPROC

```

```

10 REM::MISSILE::NEIL B. 3/85
20 MODE0
30 *K.0 RUNIM
40 g=9.805
50 speed%=100: REM = VELOCITY FOR G
GRAPH DRAWING
60 scale=10000/speed%2
70 REPEAT
80 INPUT"Enter muzzle velocity in
metres/second",V
90 UNTIL V>0
100 PRINT"SPC8"ANGLE"SPC10"RANGE"SPC
14"HEIGHT"SPC12"TIME"
110
120 FOR AX%=10 TO 90 STEP5
130   T%=10
140   PRINT AX%;" ";
150   T%=420210
160   time=FNT(V)
170   PRINT FNR(V,time),FNH(V),time
180   PROCgraph
190   SOUND0,-12,6,1
200   NEXT
210 END
220
230 DEFPROCgraph
240 LOCAL time
250 HEIGHT=FNH(speed%)
260 time=FNT(speed%)
270 MOVE 0,0
280 t%=0
290 REPEAT
300   range=FNR(speed%,t%)
310   height=HEIGHT-(g/2*ABS(time/2-
t%2))
320   DRAW range*scale,height*scale
330   t%=t%+1
340   UNTIL height<0 AND t%>1
350 ENDPROC
360
370 DEF FNH(vel)=vel2*SIN(RAD(AX%))2
2/(2*g)
380 DEF FNT(vel)=vel2*SIN(RAD(AX%))/
g
390 DEF FNR(vel,T)=vel*COS(RAD(AX%))*
T
10 REM::TWINKLE::CREATIVE COMPUTING
OCT 84
20 REM ADAPTED TO BBC BY NEIL B 3/8
5
30 REM VIEW THIS IN A DARK ROOM WIT
H REDUCED SCREEN BRIGHTNESS
40 *K.0 RUNIM
50 MODE1
60 VDU23;8202;0;0;0;
70 VDU19,2,2,0,0,0
80 VDU19,3,7,0,0,0
90 DIM AX(100),AY(100),AZ(100)
100 AP=1
110 star=60
120 TT=0
130 REPEAT
140   REPEAT
150     GCOL0,0
160     PLOT69,AX(AP),AY(AP)
170     X=RND(1279):AX(AP)=X
180     Y=RND(1023):AY(AP)=Y
190     Z=RND(3):AZ(AP)=Z
200     GCOL0,2
210     PLOT69,X,Y
220     AP=AP+1
230     IF AP>star AP=1:TT=1
240     UNTIL TT=1
250   FOR LOOP=1TO4
260     FOR TW=1 TO star
270       GCOL0,0
280       PLOT69,AX(TW),AY(TW)
290       IF INKEY(3)
300         GCOL0,AZ(TW)
310         PLOT69,AX(TW),AY(TW)
320         NEXT
330       NEXT
340     UNTIL FALSE

```

MUSIC 500 SYNTHESISER

Reviewed by Anne F.

In a wardrobe is our stereo. Twenty feet away, in a bedroom, is our computer. The connection? A 30-foot length of cable with a din plug at one end going in to the amplifier in the wardrobe, and at the other end in to a black box labelled "Music 500 synthesiser".

Aha! That still doesn't connect the stereo with the computer sez you.

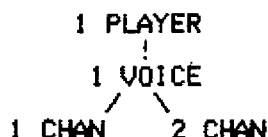
Oh yes it does - the "black box" (coloured cream) has another cable coming out of it disappearing in to the 1MHz bus under the Beeb.

Whitcoulls lent us the synthesiser plus manual plus tape that come with it, to review. One hectic week and weekend later, this is what I can tell you:

The sound quality is superb. There are several demonstration pieces ranging from Bach through Scott Joplin, 'Popcorn', to an electronic piece that is full of swirling sound (not really tuneful music). Stereo effects are easily obtained.

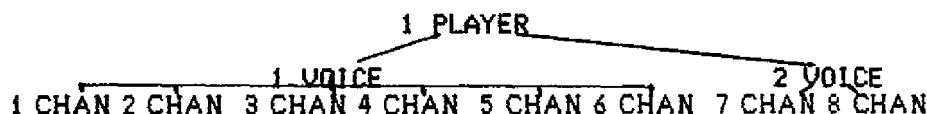
Now in to the User guide. Fairly easy to get simple melodies going. The SCAN command lets you experiment in immediate mode with the range of waves, varying pitch, amplitude, ADSR components, stereo effects or balances...

What next? You can have up to 16 different sound types playing at once - a major improvement over the Beeb's 4 or Elk's 1. There is provision for 8 players, 8 voices, and up to 16 channels. On starting up, the default is one player with one voice which has two channels:



Other players can have different voices and channels provided

the maximums above aren't exceeded. For example, Player 1 might be a "violin". Two voices could be set up: one for bowing, one for pizzicato. The bowing voice might in turn use 6 channels (channels are used in pairs usually): 1 & 2 for 'ordinary', 3 & 4 double stopping, 5 & 6 to make higher harmonics dominant.



or approximately:



In Beeb Basic the ENvelope command has 14 parameters. They set up two parts: the pitch section (the PIs and PNs), and the ADSR section (Attack, Decay, Sustain, Release). The Music 500 treats these as separate entities. To alter any part of the ADSR for instance does not mean re-defining the whole envelope, but just changing the one number:

eg if Delay already took 48 centiseconds,
20 DELAY would change it to 20 centiseconds.

There is a third 'envelope' called WAVE. This is used to set the 'impact' of the harmonics that occur when a note is played, to give the 'instrument' its distinctive timbre. A flute is regarded as a particularly pure sound - that is, when a note is played (the fundamental), only the next harmonic is evident. With a clarinet on the other hand, all the odd-numbered harmonics are evident. A good initial starting point for finding out about harmonics and waves is not the Music 500's manual, but Percy Scholes "Oxford Companion to Music" under Acoustics.

If it wasn't for the manual, I could suggest the world was your oyster when it came to synthesising music. Because it is an electronic device, you cannot hope to accurately represent "normal" musical instruments, but can invent your own distinct instruments (eg glockentock, viopling, contrabass seagull...)

Alas, the manual isn't up to standard: it is far too concise, expecting a good understanding of technical aspects of sound, and of FORTH-like linguistics. It appears to have been hastily produced: dot matrix text, with a faint ribbon; examples left

out; typing errors; a page 37A and 37B; instructions "Insert diagram here" left in; examples often unclear, with code and REMs intermingled - due thought would have led to the one line of code plus remark, new line for the next command display. The programming language is called "AMPLE". The list of command words has far too brief and highly technically worded explanations. I have studied music, play an instrument, have learned LOGO, Pascal and dabbled in FORTH. I had problems with this manual.

AMPLE is a FORTH-like language (hidden in 2 16K ROMs in the "black box").

For comparison, how to add up:

BASIC	PRINT 2+3+4	gives 9
FORTH	2 3 4 +	gives 9
AMPLE	2 3 4 #+	gives 9

By and large AMPLE doesn't use the same command words for an action as FORTH, though they work in the same way. The obvious thing about AMPLE is that you do things seemingly backwards: to set pitch for instance:

300 PITCH not PITCH = 300 or PITCH 300

It has the facility to define 'words' - roughly equivalent to PROCs (but closer to Pascal & Logo words):

"setup" [piano EMOD mpiano perc EMOD mperc square WMOD msquare
"dong" [ADSR 100 DECAY 90 SUSRTAIN]

GRAPHICS?

It seems you could produce graphics, even animated graphics, to execute whilst music is playing - a sort of 'son et lumiere' - in teletext at least. Machine code routines can be called too. I found that there appears to be a memory limitation - fairly short tunes would run in Mode 7 but not in Mode 6 (or, 6 MODE) for instance.

WHY PAY \$800 FOR THIS?

1. Because the more things plugged into your Beeb the happier you are. In which case you will especially appreciate the extra 1MHz socket at the back of the synthesiser!
2. Because you are a serious composer.
3. To experiment with 'electronic' music.
4. To be able to save your experiments on tape or disc to replay, modify, combine with other music pieces.
5. You can type in musical scores pieces (up to octets practically), pieces that you can't get on phono records to listen to...

This is what the melody line of "Doh a deer" looks like:

18,C//DE//cE/c/E///d//EFFedF/////

e//FG//eG/e/G///f//GAAGfA////

g//cDEFGA//dEFGAB////

B//eFGABC/////

24,Cb 18,a/f/B/g/C/////

ACCESS RADIO

2YB Wellington, 783KHz, has been broadcasting computer programs at 9-00am on Saturday mornings. A musical jingle will in future be played before the software transmission, at the same constant level and reference as the data following, to assist in setting up. BBC/Electron broadcast dates are:

May 18	June 15	July 20	August 17	September 21
October 19	November 16	December 21		

STOLEN BBC

Mark Patterson's BBC has been stolen in Wellington. If you hear of any machines going cheap, or without a model number, please let the Editor know (861-662(h), 729-929(wk from 20/5)), or contact the Police.

CHEAP DISCS ?

Good quality SS/DD (Single Sided/Double Density), or even SS/SD, discs are being used by some people with single density DFS (eg. Acorn) and dual sided drives, because they are cheaper than DS/DD discs. All discs have two surfaces. I understand that the difference between double or single sided, double or single density discs is essentially the manufacturer's level of guarantee, presumably reflecting quality control/testing. Thus some Single Density discs may perform quite reliably in DD systems. You pay your money and take your chance.

RADIO HAMS

The following member callsigns were extracted from the member database:

ZL4LI, ZL1BO1, ZL1BOY, ZL1BQC, ZL1BWG, ZL1TKW, ZL1TSU, ZL1UGE,
ZL1AGS, ZL1BM1, ZL1TXB, ZL2AMX, ZL2TML, ZL3ADD, ZL4NV, ZL4PE,
ZL4RP.

THE BBC MICRO IN VIDEO PRODUCTION

by Paul Davidson, CAPCOM.

CAPCOM is a small video production house specialising in training, educational and promotional programmes. We use the Beeb for a wide range of different tasks. VIEW is extremely powerful for script development and the subsequent amendments as a project proceeds. However the main use of the micro is for computer - generated graphics of all kinds for incorporation along with appropriate narration, music, sound effects etc., into video programmes.

SOFTWARE

Because the essence of television graphics is MOVEMENT, or sequences of clearly defined movements, I have not found any commercial graphics packages to be particularly useful. This is because they generally are concerned with producing single images, rather than a controllable sequence of images. I generally thus have to develop our own software, much of which is especially designed for the particular video programme in which it will be used. These are nearly always animations of varying complexity. Wherever possible I use real-time animated sequences generated by the computer. For these a good deal of colour cycling is used, generally in Mode 2 so that the greatest number of colours are available to cycle. Much of the low resolution of this mode can be overcome by judicious use of Sprites. The Graphics Extension ROM rather cunningly lets you use Hi-res sprites in Mode 2 and I have been able to do some fairly remarkable animations this way. If the micro is too slow or the animated sequence too complicated to do in real time we can of course simply edit the computer pictures together using our video editing equipment.

So over the last eighteen months we have put together a lot of interesting sequences. I have one programme which simulates a video test generator and produces a range of useful test signals, colour bars etc. Another programme generates pages of text for sub-titles, captions and credit sequences. We have a whole range of different animated graphs charts and diagrams, extremely effective in the more technical types of video production. These cover such varied topics

as the physiology of asthma, vehicle performance and sales figures, inflation rates, data transmission systems, blood pressure measurement, adventure holidays and the Michael Fowler Centre!

HARDWARE

A major problem with computer graphics is transferring the RGB images to videotape. The video signal must be broadcast standard PAL video, and the Beeb video output is anything but that!

Also in a professional studio it is necessary to somehow synchronise the computer output to the master sync pulse generator. Only then can the computer pictures be mixed, keyed or superimposed onto video from other sources, e.g. a camera or pre-recorded videotape.

At CAPCOM the computer is locked to the studio SPG by a special gen-lock unit hard-wired to the internal clock and 6845 video controller chip. A separate PAL encoder converts the RGB output to standard video and feeds it to the studio vision mixer. Once here it can be treated just like any other video input and mixed, frozen or digitally processed by any of our sophisticated video facilities. The end result is limited only by one's imagination!

THE FUTURE - HELP WANTED !

The range of immediate and future applications for the BBC at CAPCOM is so vast that one person could not pursue them all, even working full-time. If there are any competent Beeb programmers interested in video graphics applications like those discussed above, I would be glad to hear from them with a view to joint development of programmes.

Among our immediate urgent requirements are:

1. **SCROLLER:** A really smooth vertical hardware scroll for credit sequences. The programme should be able to input and store several pages of credits then "roll" them up screen as required.

2. **FONTS:** The standard Beeb typefaces are not particularly good for the bold graphics generally required on television. A machine code programme which diverts the PRINT vector and provides a choice of fonts is required. It should ideally provide the equivalent of standard printers' fonts like Helvetica Medium, Baskerville Bold etc. Many thanks to Lloyd van der Krogh for his "ITALIX" which is at least partway there. Any ideas Lloyd?

3. **TELETEXT EDITOR:** A really good teletext editor would be

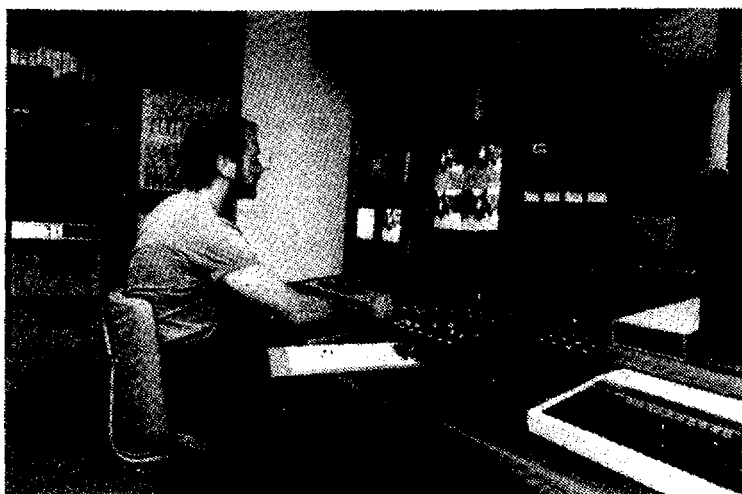
most useful. Does anyone know of one, or is anyone prepared to type in, test and debug any of those in recent literature for a modest fee?

4. TIMECODER: I have written a BASIC programme for counting hours, minutes, seconds and frames for superimposing on to prerecorded video. I would like a faster machine-code version which will output the code in real time via the cassette or RS432 port. It should also be able to read and display pre-recorded code inputted at the same port.

5. GRAPH PLOTTER: A flexible and versatile animated graph plotter is required. Should be capable of storing a series of timed "events", each event then being called up in timed sequence. "Events" might be such things as "Draw axes", "Draw 3-D Column from Y1 to Y2", "Display Sprite N% at position X%,Y%" and so on.

If anyone is interested in discussing the possibilities I would be pleased to hear from you in the immediate future.

Paul Davidson
CAPCOM
PO Box 15-175, Miramar
Wellington. PH. (04)881-121



BBC & ELECTRON SPECIALISTS
EINSTEIN SCIENTIFIC LTD & THE COMPUTER
EXPERIENCE

P.O.Box 27138
WELLINGTON
Telephone 851-774

CHECK EINSTEIN'S EXTENSIVE RANGE OF PRINTERS
MONITORS AND SOFTWARE FOR THE BBC & ELECTRON MICROS

EXCLUSIVE SPECIAL

Verbatim VALUELIFE SS/DD DISCS \$43-95 PER BOX OF 10.

EINSTEINS:

WELLINGTON

177 Willis Street, Phone Myra or Malcolm 851-855

PALMERSTON NORTH

154 Broadway Avenue, Phone Alan Carter 64-188

NEW PLYMOUTH

Corner of King and Egmont Streets, Phone Bill or Dean 85-528

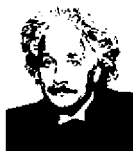
THE COMPUTER EXPERIENCE:

CHRISTCHURCH

Shop 41, 1st Floor, Cashfields Mall, Phone Garry 66-442

HAMILTON

inside D.I.C., Garden Place, Phone Ian Potts 81-969



Einstein
SCIENTIFIC LTD

The computer world made simple.

PLEASE SEND ME YOUR FREE CATALOGUE:

NAME

ADDRESS

.....
.....
.....
.....

Einstein Scientific and The Computer Experience

present The Great RITEMAN PLUS

Printer Sale

SALE PRICE \$695

SAVE \$250



As advertised in 'COMPUTE'S GAZETTE' the 'RITEMAN PLUS' is the choice of people who can't afford to say "Price is no object."

Whether you own a Commodore 64, Atari, Spectravideo, Apple, TRS80 or IBM the RITEMAN is an excellent choice. At our special sale price of \$695 we ask you to compare the RITEMAN PLUS with other printers that you have considered for your computer.

If you've been looking for a rugged, versatile, dot-matrix printer, you're probably confused by the variety of prices, quality claims and specifications quoted by each manufacturer. When you've narrowed down your choices, here are a few hints to help you decide which printer is best for you.

- How many characters per second will it print?
- Does it run continuous as well as cut sheet?
- Will it print italics, underline, and run in a compressed 132 column mode?

- How much does the ribbon cost?
- How does the price compare with the competition?

Check the features. We think you'll find that there is really no comparison... Riteman. Everything you ever wanted in a printer... for less. We have only 70 printers nationwide at the special price of \$695 and they won't last long at this price. Terms: Cash, Cheque, Bankcard, Visa and H.P. terms are available.

SPECIFICATIONS

Print Method

Serial impact dot matrix, 9 x 9.

Print Head

9 wires (user replaceable).

Print head life expectancy: 100 million characters. A 100% duty cycle (the capability to run continuously).

Print Speed

Correspondence quality printing 120 cps uni or bi-directional.

Logic seeking.

Graphic Bit Image

Standard horizontal density of 480 dots, double density of 960 dots per line.

Continuous Underline

You can underline with just one pass of the print head instead of two.

Paper Feed

a. Adjustable snap-on tractor feed (10cm - 25.5cm) (optional).

b. Friction feed (10cm - 25.5cm).

c. Pin platen (24 cm).

You can use cut sheets, fan fold or pin feed paper.

Character Set

Full 96-character ASCII with true lower case descenders. 96 italic characters. 32 block graphic characters.

Printing Mode

- a. Standard.
- b. Double Strike.
- c. Emphasised.
- d. Double Emphasized.
- e. Italics.
- f. Superscripts and Subscripts.
- g. Expanded.
- h. Compressed.
- i. Compressed/expanded.

Tab

Horizontal tab to 28 positions per line.

Buffer Size

- a. in case of pica-80 characters.
- b. in case of compressed — 132.
- c. in case of expanded — 40.
- d. Bit graphics image mode — 40.

Interface

Centronics 8 bit parallel (standard).

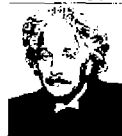
Serial Interface (Option)

300-900 baud. Easy access to dip switches.

Ink Ribbon

Reinking compact black ribbon cassette. Easily replaceable and low cost. Ribbon life expectancy: 1,000,000 characters.

Einstein
SCIENTIFIC



The computer world
made simple.

EINSTEIN SCIENTIFIC
177 Willis Street
WELLINGTON
Phone 844-353

EINSTEIN SCIENTIFIC
154 Broadway Avenue
PALMERSTON NORTH
Phone 64-198

THE COMPUTER EXPERIENCE
D.I.C. Store
Garden Place
HAMILTON
Phone 81-949

EINSTEIN SCIENTIFIC
Cnr King & Egmont Sts
NEW PLYMOUTH
Phone 85-528

THE COMPUTER EXPERIENCE
D.I.C. Store
Shop 41, 1st Floor
Cashfields Mall, Cashel St.
CHRISTCHURCH
Phone 66-442

THE COMPUTER EXPERIENCE
James Smith Store
Cuba Street
WELLINGTON
Phone 736-777



GETTING IFFY?

Anne F

It took me two years to get the hang of using DATA statements! (give or take a month) DATA files are still a mystery. My programs used to look like this:

```
10 REPEAT
20   CLS
30   food% = RND(6)
40   IF food% = 1 THEN like$ = "ORANGES"
50   IF food% = 2 THEN like$ = "ICECREAM"
60   IF food% = 3 THEN like$ = "CHOCOLATE"
70   IF food% = 4 THEN like$ = "CHEESE"
80   IF food% = 5 THEN like$ = "BLACK PUDDING"
90   IF food% = 6 THEN like$ = "COFFEE"
100  INPUT "What food do you like? "answer$
110  PRINT "I like ";like$;
120  IF like$ = answer$ THEN PRINT " too." ELSE PRINT
130  wait = INKEY(100)
140  UNTIL FALSE
150 END
```

This works beautifully. As my programs got up to 12 IFs, I threw them out of the main line of the program, into a PROC of their own.

```
40 PROCchoose_food
DELETE 50,100
170 DEF PROCchoose_food
180 food% = RND(12)
190 IF food% = 1 THEN like$ = "ORANGES"
..... and so on
320 IF food% = 12 THEN like$ = "PEACHES"
330 ENDPROC
```

It was very easy to follow through the program, and quite obvious - blatantly so? - what is going on. But perhaps starting to look a little cumbersome? All those IFs, which the computer has to work through every time, and you can no longer see the listing all on the screen at once.

This is where DATA statements can come in to their own. All the possible like\$s get tucked in to DATA lines, and the computer is told to READ through the data until it reaches the item it wants to use. As the program is written above the computer has to work through every IF statement, checking each one to see if it is what it wants,

before going on with the rest of the program. Some of this work can be cut out by using DATA.

There are other advantages to DATA:

It is easier to change; because it is usually put at the end of a program, you don't have much problem finding extra line numbers.

The 'logic' of the program is separated from the data - the same program could be used to find out what plants the computer has in its garden; but the DATA would need to be completely rewritten for instance.

This is how the first listing above would end up:

```
10 REPEAT
20   CLS
30   food% = RND(6)
40   FOR I% = 1 TO food%
50     READ like$
60   NEXT I%
70   INPUT "What food do you like? "answer$
80   PRINT "'I like ";like$;
90   IF like$ = answer$ THEN PRINT " too." ELSE PRINT
100  wait = INKEY(100)
110  UNTIL FALSE
120 END
130 DATA ORANGES
140 DATA ICECREAM
150 DATA CHOCOLATE
160 DATA CHEESE
170 DATA BLACK PUDDING
180 DATA COFFEE
```

In fact, lines 130 to 180 can be combined into one line:

```
130 DATA ORANGES, ICECREAM, CHOCOLATE, CHEESE, BLACK
PUDDING, COFFEE
```

Each word, or item of data, is separated by a comma.

Now the computer keeps reading DATA items until it has read up to food%:

eg food% = 3

It will read 3 items :ORANGES, ICECREAM, CHOCOLATE

Having reached the third item, it uses this as the definition of like\$. It stops at that point, and does not read through the rest of the data the way it worked through all the IF statements.

You have now lightened the computer's workload, so take care it doesn't go out on strike for higher pay because it is now using

"advanced technology"!!

There is one thing more to add to the last program above. If you RUN it as it stands, find out the Error message after one or more trips through the entire program.

The program works through DATA, finds the one it wants, then returns to the rest of the program. It has used the first (eg 3) items, so as far as it is concerned, these items no longer exist:

food% = 3: the computer reads item one - ORANGES - and discards it, it treats the second item similarly, reads the third item, says "aha, this is what I want", and uses it up comparing it with answer\$ in a few lines time. Next time it looks for a data item, it starts at what you would think is item 4 - CHEESE.

Very quickly the computer gives up, falls over its stomach (!) and says

'Out of DATA' to you. There is a command to insert in the program to get around this hiccough/burp:

RESTORE

This means: go back to the very first DATA statement, and start from there. or: Restore all the data previously used up. Add this line:

35 RESTORE

I said above, that a comma was used as a separator, to mark where one item of data ended and another started. What if you want a comma within an item of data? eg:

BACON,EGGS AND CHIPS

As it is, the computer would think bacon was separate from eggs and chips, but you want them to be one item. The answer is to put the whole phrase inside double quotes (speech marks)

"BACON,EGGS AND CHIPS"

Your new line of data may read:

250 DATA CHEESE,TOAST,"BACON, EGGS AND CHIPS",MARMALADE

By the way, if you add more DATA, increase the RND number to choose from. If you added three items, change RND(6) to RND(9):

30 food% = RND(9)

otherwise the computer will not know there are more than 6 items to choose from.

FILE DUMP UTILITY

Steve W

Files on the BBC generally come in two forms. The most common being files produced using SAVE, *SAVE or a series of BPUTs. Such files may be BASIC programs, machine code programs or just a block of data.

The second type of file is one produced using a series of PRINT# commands. In this type of file, the presence of fields is detected by so-called "field sentinels" which are bytes in the file with particular values which are interpreted as meaning that a particular type of field follows. Three types of field are possible:

Integer, Real, and String

For String variables, two sentinel bytes are used, with the second denoting the length of the string. BASIC needs this information as it is possible for that string to contain bytes within it which could have the same value as a field sentinel. The DFS provides a *DUMP command which will do a straight character dump of a file, but there is no inbuilt command to produce a field by a field dump.

Hence this utility program!

Within type two files, several situations are possible:

1. A collection of fields in logical groups (records)
2. Packed records.

An example of 1:

may be a file of customers where for each customer the following items of data are recorded:

Customer number	Integer
Name	String
address line 1	String
address line 2	String
address line 3	String
phone number	String
value of current accounts	Real

In this case, data within the file would probably consist of an Integer, followed by 5 strings, followed by a real number. Note that if the file contained data for more than one customer, this sequence of fields would occur as a repeating group. This is called a "logical record". Because string fields can be any length, there is (usually) no easy way to access a particular customer directly. A file of this nature is called a sequential file because access to it is always sequential in nature ie start at the beginning and read forwards.

An example of 2:

"packed" refers to the compression of information into a smaller space. Using the example above, a typical record would take up space as follows:

customer number	Integer	4+1 = 5 bytes
Name	String	30+2 = 32 bytes
address line 1	String	30+2 = 32 bytes
address line 2	String	30+2 = 32 bytes
address line 3	String	30+2 = 32 bytes
phone number	String	10+2 = 12 bytes
Value of current accounts	Real	5+1 = 6 bytes
		<u>151 bytes</u>

For a packed record, space used can be cut down:

Customer number

This is always positive, so can be fit into 2 bytes by the following method:

`cust$ = CHR$(cust% MOD 256) + CHR$(cust% DIV 256)`

This is called "packing"

To "unpack" the field:

`cust% = (ASC(LEFT$(cust$,1))*256) + (ASC(RIGHT$(cust$),1))`

For signed integers (ie may be negative), allow one extra character for a signed field.

Name and Address fields:

30 bytes each (ie no 2-byte overhead)

Value of current accounts:

Represent as an Integer, in cents. ie 31.56 becomes 3156. Then pack as for customer number.

The TOTAL length of the record string will be:

$(2+30+30+30+30+10+2) = 134 + 2 \text{ bytes overhead} = 136 \text{ bytes}$

In fact the customer number may be omitted, since if we equate customer number with record number in the file, then by setting PTR# to be: $(\text{customer number} - 1) * (\text{record length} + 2)$

we will know we are reading the correct record. This gives us random access to any customer in the file. When reading and writing records, unpacking and packing routines will be required. The saving in disc space can become significant if a large number of short fields are involved, but the information cannot easily be examined, because it may be contained within "unprintable" strings.

1. Raw data
2. Fields
3. Packed

2. Fields are detected and printed. If the number of fields per record is entered, output will be grouped into records.

3. The program will assume that the file is a series of string fields, with each string corresponding to a single record. Optionally an "unpacking" string may be used. This is created using a separate procedure which puts the unpacking string into a file called "S.PACK".

The unpacking string is a series of control characters which describe how the field is packed.

eg "X20/I2.2/F10/X10" would indicate:

a 20 character field, + a 2 byte packed integer with 2 decimal places, + a filler of 10 bytes (ie ignore next 10 bytes), + a 10 byte string.

NOTE:

Any file can be dumped as a character dump; files of packed strings can be dumped in any of the three formats. Other combinations cause the program to fail. (Try dumping a BASIC program file as a field dump!)

```

10 REM DUMP UTILITY
20 REM STEVE WILLIAMS
21 REM 61 DISRAELI STREET
22 REM EPSOM
23 REM AUCKLAND
24 REM NEW ZEALAND
30 REM 14/03/85
40 REM
50 MODE 7
60 ON ERROR GOTO 1000
70 PROCinit
80 REPEAT
90  PROCparam
100 IF format%=80 THEN MODE 3
110 PROCdump
120 MODE 7
130 PROCheading
140 PROCagain
150 UNTIL againreply%=FALSE
160 PROCend
170 END

999 REM*****
*
1000 VDU 3
1010 VDU 15
1020 CLOSE#0
1030 PRINT "Error ";ERR;" at line ";ER
L
1040 PRINT
1050 REPORT
1060 END
1499 REM*****
1500 DEFPROCinit
1590 ENDPROC
1999 REM*****
2000 DEFPROCparam
2010 CLS
2020 PROCheading
2030 REPEAT
2040 PROCinput(0,5,"(C)haracter,(F)i
eld or (P)acked (C)")
2050 IF input$="" THEN input$="C"

```

```

2060 UNTIL input$="C" OR input$="F"
      OR input$="P"
2070 type$=input$
2080 REM IF type$="P" THEN PROCpackstri
ng
2090 REPEAT
2100 PROCinput(0,7,"Output to (P)rin
ter or (T)terminal (T) ")
2110 IF input$="" THEN input$="T"
2120 UNTIL input$="T" OR input$="P"
2130 IF input$="P" THEN printer%:=TRUE
ELSE printer%:=FALSE
2140 IF printer%:=TRUE THEN format%:=80
ELSE PROCinput(0,9,"(40) or (80)
column format (40) ");IF input$="
" THEN format%:=40 ELSE format%:=VA
L(input$)
2150 PROCinput(0,11,"Enter file specif
ication ")
2160 filename$=input$
2170 ENDPROC
2499 REM*****
2500 DEFPROCdump
2510 F%:=OPENIN(filename$)
2520 IF F%:=0 THEN PROCerror("File not
found")
2530 IF printer%:=TRUE THEN VDU 2 ELSE
VDU 14
2550 PRINT "File specification: ";fil
ename$
2560 IF type$="P" THEN PRINT "Packed d
ump";PROCpackdump
2570 IF type$="C" THEN PRINT "Characte
r Dump";PROCchardump
2580 IF type$="F" THEN PRINT "Field du
mp";PROCfielddump
2590 CLOSE#F%
2600 VDU 3:VDU 15
2610 ENDPROC
2999 REM*****
3000 DEFPROCchardump
3010 REPEAT
3015 text$=""
3020 FOR sub%=1 TO format%/5
3030 IF EOF#F% THEN char$=" " :c
har%:=ASC("*") ELSE char%:=BGET
#F%:char$=STR$(char%):char$=S
TRING$(3-LEN(char$)," ")+char
$+" "
3040 IF char%>31 AND char%<127 THE
N text$=text$+CHR$(char%) EL
E text$=text$+"."
3050 PRINT char$;
3060 NEXT
3070 PRINT text$;
3080 IF printer%:=TRUE THEN VDU 1,10
3090 UNTIL EOF#F%
3091 VDU 3:VDU 15
3095 PROCcontinue
3100 ENDPROC
3499 REM*****
3500 DEFPROCheading
3510 PRINTTAB(10,2);"FILE DUMP UTILITY
"
3520 PRINTTAB(10,3);"=====
"
3530 ENDPROC
3999 REM*****
4000 DEFPROCfielddump
4005 PROCinput(0,17,"Enter fields per
record (0-?) (0)*")
4006 IF input$="" THEN fields%:=0 ELSE
fields%:=VAL(input$)
4007 records%:=0
4008 fsub%:=fields%
4010 REPEAT
4015 fsub%:=fsub%+1
4016 IF fsub%>fields% AND fields%<>0
THEN records%:=records%+1:PRINT
:PRINT "Record ";records%:PRINT
:fsub%:=1
4020 sub%:=PTR#F%
4030 A%:=BGET#F%
4040 IF A%:=0 THEN PTR#F%:=sub%:INPUT#
F%,string$:PRINT "String:(";STR
$(LEN(string$));") : ";PROCpri
ntstring:PRINT:GOTO 4100
4050 IF A%:=&40 THEN PTR#F%:=sub%:INPU
T#F%,integer%:PRINT "Integer :
";integer%:GOTO 4100
4060 IF A%:=&FF THEN PTR#F%:=sub%:INPU
T#F%,real:PRINT "Real : ";real:
GOTO 4100
4070 PRINT "Unidentified data, ASCII
value = ";A%
4100 UNTIL EOF#F%
4105 PROCcontinue
4110 ENDPROC

```

```

4499 REM*****
4500 DEFPROCinput(X%,Y%,string$)
4510 PRINTTAB(X%,Y%);STRING$(40-X%, " "
)
4520 PRINTTAB(X%,Y%);string$
4530 INPUTTAB(X%+LEN(string$)+1,Y%) in
put$
4540 ENDPROC
4999 REM*****

5000 DEFPROCerror
5010 VDU 3
5020 VDU 15
5030 PRINT "An error has occurred"
5040 VDU 7
5050 PRINT string$
5060 ENDPROC
5499 REM*****
5500 DEFPROCcontinue
5505 VDU 3:VDU 15
5510 PRINT "Press any key to continue
"

5520 A=GET
5530 ENDPROC
5999 REM*****
6000 DEFPROCagain
6010 PROCinput(0,7,"Again Y/N (Y)")
6020 IF input$="" THEN input$="Y"
6030 IF input$="Y" THEN againreply%=TR
UE ELSE againreply%=FALSE
6040 ENDPROC
6499 REM*****
6500 DEFPROCend
6560 ENDPROC
6999 REM*****

7000 DEFPROCpackdump
7005 record%/=0
7010 REPEAT
7020 INPUT#F%,string$
7030 record%+=record%+1
7040 PRINT "Record number ";record%;
" (length=";LEN(string$);")"
7045 sub%/=0
7050 FOR X%=1 TO format%/5
7055 sub%=sub%+1
7056 IF sub%>LEN(string$) THEN cha
r$=" " ELSE char$=MID$(string
$,sub%,1)
7070 IF ASC(char$)>31 AND ASC(char
$)<127 THEN PRINT char$;" "
; ELSE PRINT ".";" ";
7080 NEXT
7090 PRINT
7100 sub%=sub%-(format%/5)
7110 FOR X%=1 TO format%/5
7120 sub%=sub%+1
7130 IF sub%>LEN(string$) char$="
" ELSE char$=STR$(ASC(MID$
(string$,sub%,1)))
7135 IF LEN(char$)<4 THEN char$=ch
ar$+STRING$(4-LEN(char$)," ")
7140 PRINT char$;
7150 NEXT
7155 PRINT
7160 IF sub%>LEN(string$) THEN GOTO
7050
7170 UNTIL EOF#F%
7180 PROCcontinue
7190 ENDPROC
7400 REM*****
7500 DEFPROCprintstring
7510 FOR Y%=1 TO LEN(string$)
7520 A$=MID$(string$,Y%,1)
7530 IF ASC(A$)>31 AND ASC(A$)<127 T
HEN PRINT A$; ELSE PRINT ".";
7540 NEXT
7550 ENDPROC

```

MEMORIES ARE MADE OF THESE

David Fung

In THANKS FOR THE MEMORY (Beeblet Vol 2, No.11, P.73 - 77) I demonstrated the way to create an INDEX bypassing the need for STRING INFORMATION BLOCKS. The Listing did not provide the means to read the index, but I had hoped that the article might have spurred you on to develop it further. The function of an index is to keep tabs on where things are (like files, records, telephone numbers, addresses etc.). We put a collection of these things in a FILE.

Listing 2 is a programme using the INDEX concept to enable writing and reading of records in a File and to read the INDEX to see what records are in the FILE.

Firstly you will need to create an INDEX as before, ie.

*SAVE INDEX 4C20 7C00

Then you will need a FILE to recieve the records by reserving sufficient memory as you require, eg.

*SAVEDEFIL 00000 10000

would reserve 64K.(the last number is the hex. for 64K)

Then use Listing 1 to set Himem, Page, *load INDEX and CH. "IRMAKER"

Listing 1 "LEAD"

10REM. "LEAD" to chain "IRMAKER"

20HIMEM = &4D1F

30PAGE = &1400

40*LOAD INDEX

50CHAIN "IRMAKER"

PROCEDURES in Listing 2 in order of encountering:

PROCwrite_record	260 - 390
PROCname	590 - 650
PROCfile	410 - 460
PROCreturntomenu	1300 - 1340
PROCread_record	670 - 760

PROCstring	930 - 970
PROCsearch	780 - 910
PROCgetdata	990 - 1070
PROCread_index	1090 - 1220
PROCsave_index	480 - 570
PROCend	1360 - 1390

VARIABLES used:

M=&4D20 and the next byte M?1 contain the number of items written in the INDEX
 begin% address of the first item in the INDEX
 end% the last byte of the last item of the INDEX
 record% the number of records in file or the record number
 key% the address of a copy of name\$
 write% the address where a new item (name\$) is to be entered in the INDEX
 search% is the current address where an item is being searched for
 N% is the nth byte of an item being compared with the corresponding byte in the name\$
 name\$ is the name of an item being entered or looked for in the INDEX
 data\$ is the content of a record
 read% is the current address of an item being read in the INDEX
 option% being the choice of functions displayed in the MENU

flags:

end_flag to check that items written to or amended in the index are to be *SAVED to the disc "INDEX" file
 next_name item not yet found, go on to look at the next
 stopsearching searched to the last items in INDEX

The SEARCH ROUTINE:

PROCsearch

If the item (name\$) is 10 bytes long then the last byte search%?9 MUST be a printable character, if it is not then goto next item. If it is a printable character, then it may be the item looked for, so start comparing the characters of name\$ and at search% byte by byte. If all 10 characters matched then that is the item, else goto the next item (search%=search%+12).

PROCstring

If the item is less than 10 bytes then the byte following the last character must contain 13 (CR), therefore that item can be handled as a string\$, and

can be compared with name\$, if they do not match goto the next item until it is found, or the search has come to the end%.

The SEARCH as you see is a serial search. It is fine if you have not more than a few hundred items. The search can be made faster if the items are sorted in alphabetical orders, and faster still if a BINARY CHOP search is used instead. There are many other ways to sort and search and it is quite fun and instructive to try the various methods.

I have assumed each record is 256 bytes long, it is therefore relatively easy to calculate where a record is in "DEFILE" (read Steven R.'s articles on Files). You may want to make it longer or shorter, or you may want to input several data\$ / data%, so you modify PROCwrite_record to your own requirement. You may also notice that "DEFILE" as defined would not hold 1000 records. If you want to make full use of the index you would need to open several "DEFILEn". How? How about you figure it out?

```
10 MODE 7
20 PRINTTAB(10,10); "INDEX / RECORD
  / FILE MAKER"
30 WAIT = INKEY(200)
40 end_flag=1
50 ON ERROR REPORT:PRINTERL:CLOSE#0:
  END
60 REPEAT
70   CLS
80   M=&4D20:begin%=&4D22:key%=&C01
90   record% =?M?M?1*256:end%=&4D22+
    (record%*12)
100  PRINTTAB(10,5) "1. WRITE RECORD
    "
110  PRINTTAB(10,7) "2. READ RECORD"
120  PRINTTAB(10,9) "3. AMEND RECORD
    "
130  PRINTTAB(10,11) "4. READ INDEX"
140  PRINTTAB(10,13) "5. SAVE INDEX"
150  PRINTTAB(10,15) "6. END"
160  INPUT TAB(10,24) "WHICH OPTION?"
    " option%
170  CLS
180  IF option%=1 PROCwrite_record
190      IF option%=2
200      IF option%=3 PROCamend
210      IF option%=4 PROCread_index
220      IF option%=5 PROCsave_index
230      IF option%=6 PROCend
240      UNTIL option%=6:END
250  ;
260  DEF PROCwrite_record
270  IF record%#1000 THEN PRINT""TAB(
    15)"INDEX IS FULL" : ENDPROC
280  record%=record%+1
290  write%=begin%+(record%-1)*12
300  PROCname
310  INPUTLINE""DATA TO BE RECORDED "
    ,data$
320  PROCfile
330  $write%= name$ :REM. enter name$
    in index
340  write%?10 = record% MOD 256
350  write%?11 = record% DIV 256
360  ?M=write%?10
370  M?1=write%?11
380  PROCreturntomenu
390  ENDPROC
400  ;
```

```

410 DEF PROCfile
420 c=OPENUP "DEFILF"
430 PTR#c=(record%-1)*256
440 PRINT#c,data$
450 CLOSE#0
460 ENDPROC
470 :
480 DEF PROCsave_index
490 M=record% MOD 256 : M?1 = reco
rd% DIV 256
500 REPEAT
510 INPUT "DO YOU WISH TO SAVE INDE
X? "A$
520 UNTIL A$ = "Y" OR A$ = "N"
530 IF A$ = "N" ENDPROC
540 *SAVE INDEX 4020 7C00
550 end_flag=1
560 PROCreturntomenu
570 ENDPROC
580 :
590 DEF PROCname
600 REPEAT
610 PRINTTAB(38,15)"**"
620 INPUTLINE TAB(0,15)"PLEASE ENTE
R NAME OF RECORD "name$
630 IF LEN(name$) > 18 PRINT "NAME
TOO LONG":PRINTTAB(29,15)"
640 UNTIL LEN(name$) < 11 AND LEN(n
ame$) > 0
650 ENDPROC
660 :
670 DEF PROCread_record
680 N/=0:search% = begin%
690 PROCname
700 CLS
710 $key% = name$
720 IF LEN(name$) < 10 PROCstring ELSE
PROCsearch
730 IF $search% = name$ OR N/=10 PROCge
tdata
740 IF search% = end% PRINTTAB(10,12) n
ame$;" NOT IN INDEX"
750 PROCreturntomenu
760 ENDPROC
770 :
780 DEF PROCsearch
790 stopsearching=FALSE
800 REPEAT

```

```

810 REPEAT
820 IF search% < 33 OR search% > 99
26 search% = search% + 12
830 UNTIL search% > 32 AND search
% < 99 < 127 OR search% = end%
840 IF search% = end% stopsearching=T
RUE
850 REPEAT
860 next_name=FALSE
870 IF stopsearching=FALSE AND se
arch% < N/=key% < N/=N/ N/=N/+1
880 UNTIL N/=10 OR next_name=TRUE
890 IF next_name=TRUE search% = searc
h% + 12
900 UNTIL N/=10 OR stopsearching=TR
UE
910 ENDPROC
920 :
930 DEF PROCstring
940 REPEAT
950 IF $search% < name$ search% = sear
ch% + 12
960 UNTIL $search% = name$ OR search%
= end%
970 ENDPROC
980 :
990 DEF PROCgetdata
1000 record% = search% ? 10 + (search% ? 11) * 2
56
1010 c=OPENUP "DEFILF"
1020 PTR#c=(record%-1)*256
1030 INPUT#c,data$
1040 PRINTTAB(0,12)name$,TAB(15);"RECO
RD ";record%
1050 PRINT"data$
1060 CLOSE#0
1070 ENDPROC
1080 :
1090 DEF PROCread_index
1100 key% ? 10 = 13:read% = begin%
1110 VDU14
1120 REPEAT
1130 FOR N/=0 TO 9
1140 key% < N/=read% < N/=
1150 NEXT
1160 record% = read% ? 10 + (read% ? 11) * 256
1170 PRINTrecord%;" ";$key%
1180 read% = read% + 12

```

CREATIVE ASSEMBLER:

How To Write Arcade Games for the BBC Microcomputer Model B and Acorn Electron.

by Jonathan Griffiths: Published by Penguin Acorn Computer Library.

Reviewed by Warren W.

This book, by one of Acornsoft's games programmers, author of such well known games as SNAPPER and ROCKET RAID, is set out in three parts.

The first part introduces the more common and useful assembler instructions and illustrates their use by examples. Starting from basics the author discusses jumps, branches and loops, and looks at logical operations, addressing modes and the stack.

The second part deals with more complex programming techniques such as macros and operating system routines. This section is aimed at those ambitious people wanting to write large assembler programs.

The games programmer is the target for the third part. A useful and comprehensive collection of routines and utilities is followed by a chapter on graphics shape design. The book is rounded off with an example program, a simple version of Space Invaders, which illustrates many of the points covered in earlier chapters.

To my delight this book includes a chapter on program structure, which deals with many very important aspects of design which are fundamental to good programming practice. From my experience this aspect of assembly language programming is often treated much too lightly. The use of good structure is more vitally important at this level than when programming in BASIC. Also of interest is a chapter on the differences between BASIC I and BASIC II, which includes techniques for simulating the EQU assembler directives and the OSCLI command in BASIC I.

The layout of the book is excellent, with all listings in bold blue print. The book was prepared on a BBC Microcomputer using the VIEW wordprocessor - what better recommendation!

The only adverse criticism that I have is that despite a long list of people thanked for their help with proof reading, a

disappointing number of generally trivial typesetting errors have been allowed to remain in what is really a very good book.

In summary, this book will, I am sure, prove to be a source of guidance and inspiration to anyone with an elementary knowledge of assembly language programming who wants to get into the business of writing good arcade games. Isn't it a pity that the ideas for good arcade games are still hard to come by! All in all this book is good value at \$16.95. I look forward to further books from this official Acorn joint venture.

```

1180     UNTIL read/=end/
1190     VDU15
1200     PROCreturntomenu
1210     ENDPROC
1220     :
1230     DEF PROCamend
1240     PRINT"NOT INCLUDED IN THIS LISTING."
1250     PRINT"WATCH FOR NEXT INSTALMENT."
1260     PRINT"OR DO IT YOURSELF"
1270     PROCreturntomenu
1280     ENDPROC
1290     :
1300     DEF PROCreturntomenu
1310     IF option%=1 OR option%=3 end_flag=0
1320     PRINT""PRESS ANYKEY TO CONTINUE" 1330 KEYHIT=GET
1330     ENDPROC
1340     :
1350     DEF PROCend
1360     IF end_flag=0 PRINT TAB(0,10)"DO YOU MEAN NOT TO SAVE THE INDEX? (Y/N)" :I
F GET$="N" option%=0 :ENDPROC
1370     IF end_flag=0 PRINT"O.K. CHARLIE! SORRY I ASKED."
1380     ENDPROC

```

GRID

Tim R.

Well here is another procedure to add to your growing library of useful routines. The idea is that this little program will (given the correct parameters) put a grid of any size on the screen. Only, one thing to remember is that the routine will not work in modes 3,6, and 7.

I've had to cheat a little with the algorithm due to the problems of BASIC rounding numbers. I have played around with the values until I am happy that you will always get a perfect looking grid.

Hope this is of some use to those of you who are budding writers of such games as Battleships et al.

I am currently working on an assembly routine to give eight graphic and eight text screens at once. To be able to draw a grid of 4 by 4 boxes then write text and graphics into any with scrolling etc, might be very useful.

```
10 REM *** PROCEDURE TO DRAW A GRID OF ANY SIZE AND SHAPE ***
20 REM *** BY TIM RYAN ***
30
40 REM Example of use given on line 88
50 REM Coords leftX etc refer to corners of grid (as in window command)
60 REM Xboxes & Yboxes = number of posions in grid for each direction
70 CLS
80 PROCgrid(20,20,1200,1000,14,7)
90 END
100
110 DEF PROCgrid(leftX,botY,rightX,topY,Xboxes,Yboxes)
120 LOCAL across,down,Xstep,Ystep
130 Xstep=(rightX-leftX)/Xboxes+1E-7
140 Ystep=(topY-botY)/Yboxes+1E-7
150 FOR across=leftX TO rightX STEP Xstep
160   MOVE across,botY:DRAW across,topY
170   NEXT
180 MOVE rightX,botY:DRAW rightX,topY
190 FOR down=botY TO topY STEP Ystep
200   MOVE leftX,down:DRAW rightX,down
210   NEXT
220 MOVE leftX,topY:DRAW rightX,topY
230 ENDPROC
```

RE: ANNE'S COMPLAINT - Beeblet, March '85

by Geoff P.

Most of the content of Beeblet comes from members in the Wellington area. The Editor lives in the Wellington area. These two facts are not mere coincidence. Our Editor does not edit in the true sense of the word; he does not sit at a desk heaped with contributions and decide "Yes, I'll give that two pages" or "Perhaps I'll hold this over for a month", instead he has to hustle for them. I believe his job is often a matter of trying to pad out forty pages with the few contributions he receives. Don't moan then if few the people that Bruce hustles successfully happen to write about things that don't particularly interest you. The User Group has around four hundred members. Beeblet consists of forty pages with roughly fifty lines per page. If every member contributed just five lines each month we'd fill the magazine. And imagine how interesting such a magazine would be with four hundred very different people talking to one another!! With this in mind I make the following suggestion; that one member be appointed to the committee in each area with the specific responsibility for producing a certain number of pages for the magazine each month.

Basically, this will simply give us half a dozen people hustling for contributions but because of geographic separation we should get a nice wide variety of work. Based on David Fung's, (a Wellington member!), membership chart of a couple of months back I would suggest the following minimum contributions:

Auckland - 8 pages
Tauranga - 1.5 pages
Hamilton - 2.5 pages
Chch - 2.5 pages
Dunedin - 1.5 pages

As you can see Wellington will still produce over half the magazine - we do anyway so there's no problem there! What it will give us is some variety in the content. Of course the key word is commitment. It takes a little effort to produce something each month and as you can see from these pages most people can't be bothered. Moaning that things aren't exactly as they'd like is much easier. There's something ironic in the fact that in the month that Anne's complaint was published her husband, the ubiquitous GCF, (another Wellington member!), contributed over 25% of the magazine's content - eleven out of a total of forty pages - on his own!!!

" S P E C I A L T E R M S "

" S P E C I A L T E R M S "

B B C / A C O R N M I C R O C O M P U T E R U S E R S G R O U P

* JUKI 2200 ELECTRONIC TYPEWRITER/PRINTER *

NORMAL PRICE ****\$895.00****

S C H E M E 1 — JUKI 2200 INTRODUCTORY PACK
CONTAINING:
Juki 2200 Typewriter/Printer
6 Correctable Carbon Ribbons
6 Juki Lift off Tapes

\$994.00 VALUE

YOU ONLY PAY ****\$895.00****

S C H E M E 2 — JUKI 2200 PRINTER PACK
CONTAINING:
Juki 2200 Typewriter/Printer
BBC Cable Interface
4 Juki Nylon Ribbons

\$990.00 VALUE

YOU ONLY PAY ****\$895.00****

S C H E M E 3 — JUKI 2200 SOLD PACK
CONTAINING:
Juki 2200 Typewriter/Printer

\$895.00 VALUE

YOU ONLY PAY ****\$800.00****

P A Y M E N T

With Order.
Cash, Cheque or Credit Card.

LIMITED OFFER - EXPIRING 30/6/85

PLEASE COMPLETE COUPON ON THE ATTACHED COPY ADVERTISEMENT
QUOTING YOUR MEMBERSHIP NUMBER OF THE BBC/ACORN MICROCOMPUTER
USERS GROUP.

THE DIRECT WAY TO SAVE.

HALL OF FAME

PUBLISHER:	GAME	HIScore	SCORER
Acorn	Arcadians	21,470	Malcolm Mac
	Aviator	5,610	Joanne C's Dad
	Countdown	246	Andrew P.
	Elite	Elite	J&C Climo*
	Meteors	46,200	Andrew P.
	Monsters	107,390	Robert T.
	Planetoid	413,875	Peter J.
	Rocket Raid	140,777	Mark R.
	Snapper	250,000	Susan
	Starship Command	9,495	Robert N.
	Super Invader	20,420	P. Leadley
	Cylon Attack	157,970	Aaron
?	Bug Blaster	37,642	Steve W.
Alligat.	Zalaga	215,180	Peter J.
Aardvark	Frak	1,026,820	Andrew P.
	Monsters	175,110	Geof P.
Amcom	Fortress	89,900	Andrew P.
Bugbyte	Twin Kingdom	385	A. Beeston
Comp.Con	Android Attack	236,195	Peter J.
Dr Soft	'747'	205	Steve W.
FG Soft.	Mutant Defender	6,900	Steve W.
	Lords of Time	925/1000	Steve W/Kerry K.
	Snowball	900/1000	Steve W./Dianne
Prog. Pow	Felix the Factory	7,000	Dianne S.
	Frenzy	915,468	Dianne T.
	Killer Gorilla	91,000	Susan
	Laser Command	109,225	P. Leadley
	Moon Raider	112,350	P. Leadley
Sup. Soft	G*Bert	43,825	Peter J.
	Roadrunner	97,000	Andrew P.

* (disc) 2 tasks performed

HALL OF FAME

USER GROUP MEETINGS

- AUCKLAND:** -meets 2nd Wednesday of the month, 7.30pm, Conference Rooms, Auckland Secondary Teachers College, 60 Epsom Ave, Epsom. ph Dave 770-630 x 518(bus) or Kerry 695-355.
 -North Shore: meets last Wednesday of the month, 7.30pm, Salvation Army Hall, Glenfield Road, Glenfield ph Chris 419-0543 (wk)
- CHRISTCHURCH:** -fortnightly workshops, Monday 6.30pm at Hagley High School. ph. Michael 582-267.
- DUNEDIN:** -first Thursday of the month. Phone Martin McDowall 42-031
- HAMILTON:** -last Thursday of the month 4.45pm Waikato Tech B-block staff -room. Ph. Peter, Hamilton 392-500 x897(bus) or Alison, Morrinsville 6695.
- HAWKES BAY:** -Alternative months Hastings & Napier. Contact Kendall Napier 435-624, Bob Taradale 446-955, Mitch Hastings 778-235.
- INVERCARGILL:** -Phone Eyan 330-444 Invercargill.
- PALMERSTON NORTH:** -Phone Tom Skinner 83-009(wk)
- TAUPO:** -Contact Dave 84-215.
- TAURANGA:** -every second Wednesday. ph. Chris & Jane 65-076.
- TIMARU:** -Phone Lloyd van der Krogt Timaru 61-412.
- TOKOROA:** -meets first Friday of each month, Tokoroa High School, 8pm. Phone Brian 67-625 Tokoroa.
- WELLINGTON:** -meets last WEDNESDAY of the month, 7.30pm, Royal Society Lecture Theatre, Turnbull St, Thorndon. ph. Don 848-235 or Beryl 862-874 or Mike 785-437.
- OTHER CENTRES:** - let us know and we will publish details here.

THE NEWSLETTER:

CONTRIBUTIONS: most welcome. Listings should be sent in on tape or disc. (Tape - 2 copies, one at 300 baud please). A 35c stamp gets your tape / disc returned too. Please include written explanation of listing; text only material welcome too.

DEADLINES: - For a particular issue, the last day for material is the last day of the month prior to publication.

ADVERTISING: - Rates are \$20 per half page (camera ready), deadline as above.

Copyright (c) THE BBC/ACORN COMPUTER USER GROUP OF NZ INC.

Published monthly except January and mailed to financial members.

MEMBERSHIP:

Membership of the users group is on payment of an annual subscription from April to March. For the 1985/6 year it is: \$30.00, plus \$5.00 for Auckland & Wellington members as a local branch levy, which entitles attendance at meetings as detailed above. Renewing members get all back issues of the current year.

THE BBC/ACORN COMPUTER USER GROUP OF NZ INC, P.O.Box 9592, Wellington.

Reproduced by ROBERTS PRINTING Co. Ltd, Wellington.